Robust statistical methods: a primer for clinical psychology and experimental psychopathology researchers

Article (Accepted Version)

Robust statistical methods: a primer for clinical psychology and experimental psychopathology researchers

Andy P. Field[a] & Rand R. Wilcox[b]

[a]School of Psychology, University of Sussex, Falmer, Brighton, BN1 9QH, UK

[b]Department of Psychology, University of Southern California, 618 Seeley Mudd Building, University Park Campus, Los Angeles, CA 90089-1061, USA

Correspondence to: Andy P. Field, School of Psychology, University of Sussex, Falmer, Brighton, East Sussex, BN1 9QH, UK. [E-mail: andyf@sussex.ac.uk]

Abstract

This paper reviews and offers tutorials on robust statistical methods relevant to clinical and experimental psychopathology researchers. We review the assumptions of one of the most commonly applied models in this journal (the general linear model, GLM) and the effects of violating them. We then present evidence that psychological data are more likely than not to violate these assumptions. Next, we overview some methods for correcting for violations of model assumptions. The final part of the paper presents 8 tutorials of robust statistical methods using R that cover a range of variants of the GLM (*t-tests*, ANOVA, multiple regression, multilevel models, latent growth models). We conclude with recommendations that set the expectations for what methods researchers submitting to the journal should apply and what they should report.

Keywords Robust statistical methods, assumptions, bias

Robust statistical methods: a primer for clinical psychology and experimental psychopathology researchers

Overview

The general linear model (GLM), which is routinely used in clinical and experimental psychopathology research, was once thought to be robust to violations of its assumptions. However, based on hundreds of journal articles published during the last fifty years, it is well established that this view is incorrect. Moreover, modern methods for dealing with the violations of these assumptions can result in substantial gains in power as well as a deeper, more accurate and more nuanced understanding of data. We begin with an overview of the key assumptions underlying the GLM. We then review various misconceptions about how robust the GLM is to violations of those assumptions and look at the effects that violations can have. We end the first section by looking at the evidence that psychological data, in general, are likely to violate the assumptions of the GLM.

In part 2 of the paper we overview as election of ways to deal with violations of assumptions that fall under the headings of data transformation, adjustments to standard errors, and robust estimation. In the final part, we present 8 tutorials that use datasets relevant to this journal to show how to implement a selection of techniques (robust estimators for model parameters and standard errors) for designs common to this journal (comparing dependent and independent means, predicting continuous outcomes from continuous predictors and longitudinal designs).

The assumptions of the general linear model

Critical assumptions

Psychology researchers (generally) and those with interests in psychopathology (specifically) typically apply variants of the general linear model to their data. In this model, an outcome variable (*Y*) is predicted from a linear and additive combination of one or more predictor variables (*X*). For each predictor there is a parameter that is estimated from the data ($\hat{b}$) that represents the relationship between the predictor and outcome variable if the effects of other predictors in the model are held constant. There is a parameter (the constant, $\hat{b}_0$) to estimate the value of the outcome when all predictors are zero. The error in prediction is represented by the residual (ε_i), which is (for each observation, *i*) the distance between the value of the outcome predicted by the model and the value observed in the data (Eq. 1). Model parameters (the $\hat{b}$s) are typically estimated using ordinary least squares (OLS) estimation, which seeks to minimize the squared errors between the predicted and observed values of the outcome, or maximum likelihood (ML) estimation, which seeks to find the parameter values that maximise the likelihood of the observations.

$$\hat{Y}_i = \hat{b}_0 + \hat{b}_1 X_{1i} + \cdots + \hat{b}_n X_{ni} + \varepsilon_i \qquad \textit{Eq. 1}$$

It is widely known that the general linear model is a flexible framework through which to predict a continuous outcome variable from predictor variables that can be continuous (often termed as ‘regression’ or ‘multiple regression’), categorical (often referred to as ‘ANOVA’) or both (often referred to as ‘ANCOVA’). Similarly, experimental designs containing repeated measures and longitudinal data are special cases of a multilevel linear model in which observations (level 1) are nested within participants

(level 2). Despite the proliferation of terms that create artificial distinctions in the statistical models being applied, research designs that might, by many, be labelled as ‘regression’, ‘ANOVA’, ‘ANCOVA’, and multilevel models, are all variants of the linear model and, therefore, have a common set of underlying assumptions (see Cohen, 1968; Field, 2013; 2016, for tutorials).

The linear model has two main assumptions: (1) additivity and linearity, and (2) spherical residuals. The assumption of spherical errors implies that residuals are both independent and homoscedastic. This assumption is typically examined with respect to these two implications. Independent residuals are ones that are not correlated across observations. You would expect this assumption to be true when each observation comes from a different entity, but false when observations come from the same entities at different time points (e.g., longitudinal designs) or from different entities that share a context relevant to the outcome variable (e.g., clients being treated by the same clinician, or children taught by the same teacher). Correlation across residuals is known as *autocorrelation*. Homoscedastic residuals are ones that have the same variance for all observations. Residuals without this property are called *heteroscedastic*.

When using the general linear model researchers assume that Eq. 1 is a valid representation of the real-world process that they are trying to model. In short, they assume that the outcome variable is linearly related to any predictors and that the best description of the effect of several predictors is that their individual effects can be added together. As such, the assumption of additivity and linearity is the most important because it equates to the general linear model being the best description of the process of interest. If this assumption is not true then you are fitting the wrong model.

The assumptions of independent and homoscedastic residuals (i.e., spherical errors) relate to the Guass-Markov theorem, which states that when these conditions are met (and residuals have a mean of zero) then the linear model derived from OLS estimation will be a *best linear unbiased estimator*. In other words, it will be the unbiased linear estimator that has the least variance (i.e., is optimal)[i]. 'Unbiased' means that the estimator's expected value for a parameter matches the true value of that parameter. The consequence of violating either of these assumptions is the same: the parameter estimates themselves remain unbiased, but are no longer optimal (that is, you can find estimates with lower variance). Furthermore, the formula for the variance of a parameter (*b*) assumes a constant variance so under heteroscedasticity this formula is incorrect. Consequently, estimates of the standard error of the parameter (which are based on the variance) are biased (Hayes & Cai, 2007). The presence of autocorrelation biases the standard errors of model parameters too.

Biased standard errors have important consequences for significance tests and confidence intervals of model parameters. For example, the test statistic, *t,* associated with a parameter estimate in the linear model is calculated using Eq. 2, from which a *p*-value is derived. If the standard error of the parameter is incorrect, then *t* (and the associated *p*) will be biased[ii] and have poor power (Wilcox, 2010). Similarly, the bounds of a parameter estimates' confidence interval is constructed by adding or subtracting from the estimate the associated standard error multiplied by the quantile of a null distribution associated with the probability level assigned to the interval. For example, under normality and when the variance is known, and the goal is to compute a 95% confidence interval for the mean, the standard error of the sample mean is multiplied by 1.96, the 97.5 percentile of a standard

normal distribution. Therefore, if the standard error is biased, the confidence interval will be too. Confidence intervals can be "extremely inaccurate" when the homoscedasticity assumption is violated (Wilcox, 2010).

$$t_{n-p} = \frac{\hat{b}_{observed} - \hat{b}_{expected}}{SE_{\hat{b}_{observed}}} \qquad \textit{Eq. 2}$$

Normality

An additional assumption that is often discussed in relation to the linear model is normality. There are three issues related to normality, the first of which is normality of residuals (the ε_i in Eq. 1). Each case of data has a residual – the difference between the predicted and observed values of the outcome. If you inspected a histogram of these residuals for all cases, you would hope to see a normal distribution centred around 0. A residual of 0 means that the model correctly predicts the outcome value. Therefore, if the residual is zero (or close to it) for most cases, then the error in prediction is zero (or close to it) for most cases. If the model fits well, we might also expect that very extreme over- or underestimations occur rarely. A well-fitting model then would yield residuals that, like a normal distribution, are most frequent around zero and very infrequent at extreme values. This description explains what we mean by *normality of residuals*.

The Guass-Markov theorem does not assume normally-distributed residuals: even if residuals are not normally-distributed the OLS estimator will yield a model that is the best linear unbiased estimator (i.e., unbiased and optimal). In this respect, normality of residuals does not matter. If the residuals *are* normally distributed in the population, then the OLS estimator becomes the ML estimator (that is OLS and ML estimation yield identical estimates), and it will be the most accurate. That is to say, when residuals are not normally distributed, parameter estimates will be unbiased and optimal (with respect to

minimizing the variance), but there may be classes of estimator (other than OLS) that are more accurate (Wilcox, 2010).

A simple example of this point is the (arithmetic) sample mean, which is an OLS estimator for the population mean. When the residuals associated with a sample mean (i.e., the deviation between each observed value and the mean) are not normal (for example when there are outliers), the sample mean will still be the value with the least squared error — the lack of normally-distributed residuals does not affect that fact. However, there will be more accurate estimates of the centre of the distribution of scores (for example, the median or a trimmed mean). In more formal terms, trimmed means, including the median, can have substantially smaller standard errors than the mean. This result that was derived about two centuries ago by Laplace when using the median (e.g., Wilcox, 2010).

The second normality-related issue is that *p*-values associated with the parameter estimates of the model are based on the assumption that the test statistic associated with them follows a normal distribution (or some variant of it such as *t*). Essentially, to test the hypothesis that the parameter estimate is not equal to 0 (the null hypothesis) it is necessary to assume a particular shape (i.e., normal) for the null distribution of the test statistic. If the sampling distribution of the test statistic turns out not to be the assumed shape (i.e. normal) then the resulting *p*-values will be incorrect.

The final, related, issue is confidence intervals. As already mentioned, the bounds of confidence intervals for parameter estimates are constructed by adding or subtracting from the estimate the associated standard error multiplied by the quantile of a null distribution associated with the probability level assigned to the interval. For tests of parameters in the linear model, the null distribution is assumed to be normal. It is an

example of a general strategy in inferential statistics to convert an estimator, such as the mean, into a standardized statistic (Z) that is asymptotically standard normal. The general issue is one of determining under what circumstances assuming normality gives a reasonably accurate result.

A common claim, based on the central limit theorem, is that with sample sizes greater than 30 the parameter estimate will have a normal sampling distribution. The implication being that if our sample is large we need not worry about checking normality to know that confidence intervals and *p*-values for a parameter estimate will be accurate (Lumley, Diehr, Emerson, & Chen, 2002). In which case, we can effectively ignore normality in all but quite exceptional cases of fitting a linear model. However, two things were missed when arriving at this conclusion. First, the conclusion is based on work using very light-tailed distributions. Second, the assumption that Student's *T* performs well if the sample mean has, to a close approximation, a normal distribution turns out to be incorrect under general conditions (Wilcox, 2016, 2017).

Misconceptions about robustness

There is a pervasive misconception that the general linear model is robust to violations of its underlying assumptions. Let us take, as an example, the *F*-statistic, which assesses the overall fit of the model. In experimental research, in which the predictors in the linear model represent groups of people in different treatment conditions, the 'overall fit of the model' becomes a test of whether group means differ. Early work suggested that *F* controls the Type I error rate under conditions of skew when group sizes are equal (Donaldson, 1968; Glass, Peckham, & Sanders, 1972). However, more recent investigations revealed that differences in skewness, non-normality and heteroscedasticity

interact in complicated ways that impact power (Wilcox, 2017). For example, it was believed that as kurtosis increases, the Type I error rate decreases and quickly drops below its nominal .05 level, and consequently power decreases (Glass et al., 1972). We now know that this conclusion is correct *only* if distributions have the same amount of skewness, because in this situation the *difference* between variables will have a symmetric distribution.

Unequal variances (violations of homoscedasticity), have relatively little influence when group sizes are equal and the normality assumption is true, but when group sizes are unequal *F* varies in how liberal or conservative it is as a function of whether the largest group has the smallest variance or vice versa (see Field, Miles, & Field, 2012, for a review). When normality cannot be assumed equal group sizes do not save *F* from violations of homoscedasticity (Wilcox, 2010, 2016, 2017).

To sum up, under general conditions, *F* is not robust to violations of assumptions. Wilcox (2016) sums up the situation for *t* and *F* by saying that they can be considered to be robust (at least with respect to Type I error control) if the group distributions are identical (e.g., the exact amount of skew is the same across groups). In practical terms, if the *F* is significant, it is reasonable to conclude that the distributions differ in some manner. If *F* is not significant, this might be because there is little or no differences among the distributions, or it might be because the *F* test is insensitive to important differences that were missed due to violating assumptions.

It is easily demonstrated that heavy-tailed distributions can have profound effects on both power and effect sizes; even slight departures from normality can be a practical concern. Wilcox, Carlson, Azen, and Clark (2013) illustrate this by showing that when a *t*-

statistic is used to compare, at the .05 level, two normal distributions (n = 25) with variance of 1 and means of 0 and 1, the power is .9. When these distributions are changed to be mixed normal distributions (that is distributions where 90% of scores come from the standard normal and 10% come from a normal distribution with M = 0 and SD = 10) the power drops to .28 (despite changing only 10% of scores). These kinds of contaminated normal distributions also profoundly impact effect size estimates with, for the aforementioned example, Cohen's d dropping from 1 when distributions are normal to 0.28 when they are 10% contaminated (i.e. have heavier tails).

The second misconception, to which we have already alluded, is that the central limit theorem means that in samples larger than about 30, normality does not matter. While it is true that when distributions are symmetric and have light tails the sampling distribution of means is approximately normal using samples of only 20, when distributions are asymmetric (skewed), even light-tailed distributions can require sample sizes >= 200 when using the one-sample t-test. Using a homoscedastic method, when there is heteroscedasticity, makes matters worse. When distributions have heavy tails samples need to be much larger (up to 160 in some cases) before the sampling distribution is normal (Wilcox, 2010). As such, researchers can be lured into a false sense of security that they can assume normality of the sampling distribution because of the central limit theorem.

What kind of data do psychologists usually have?

All of the above matters only if researchers in clinical psychology or experimental psychopathology typically find themselves faced with data that compromise the assumptions of the models they fit. There is a compelling case that psychological data in

general, and data measuring clinical constructs specifically, are more often than not problematic.

Micceri (1989) studied the distributional characteristics of 440 large-sample psychology-relevant measures. Remarkably, when looking at tail weight only 15.2% approximated a normal distribution and nearly 67% had at least one tail that was moderately to extremely heavy. In terms of symmetry, only 28.4% approximated a normal distribution with the remainder moderately to extremely skewed. Looking at both symmetry *and* tail weight together only 6.8% of the 440 distributions approximated normality. These data show that tail weight and symmetry consistent with a normal distribution is extremely rare in psychological data. It is particularly noteworthy, given the profound impact of heavy tails on power and effect sizes, that up to two thirds of distributions had heavy tails.

Data measuring constructs relevant to clinical psychology and experimental psychopathology show similar non-normal patterns. Substantial skew has been found in economic and cost indicators (Barber & Thompson, 2000; Hlatky, Boothroyd, & Johnstone, 2002) and measures of quality of life (Arostegui, Nunez-Anton, & Quintana, 2007) and social functioning (Tyrer et al., 2005) in clinical trial data. Skew is the norm in measures of depression (Rutter & Miglioretti, 2003; Zimmerman, Chelminski, & Posternak, 2004), mania (Picardi et al., 2008) and suicidal ideation (Binks et al., 2006). Experimental psychopathology research typically measures clinical constructs in analogue populations, and these too often have heavily skewed distributions (Rutter & Miglioretti, 2003; Tyrer et al., 2005; Zimmerman et al., 2004). Although these distributional shapes do not prevent the parameter estimates of the linear model derived from OLS from being

unbiased and optimal, other classes of estimator may be more accurate. Also, these distributions cast doubt on the validity of assuming the normal sampling distributions required by the significance tests associated with linear models.

Solutions to violated assumptions

Transforming data

A common approach to non-normality (especially skew) is to transform the data using a mathematical function such as the log or square root that decreases large values more than small ones, therefore, compressing the tail of the distribution. However, transformations are not a panacea for non-normality for several reasons. (1) Glass et al. (1972) conclude that transformations are seldom worth the effort because their potential to improve the validity of probability statements is low; (2) transforming changes the hypothesis being tested (for example, if you compare the means of log transformed variables you are comparing geometric, rather than arithmetic, means); (3) transformations muddy the interpretation because transforming the data also transforms the construct that it measures (Grayson, 2004); (4) for a transformation to have any benefit it must be clear that the consequences of applying the 'wrong' transformation are less severe than the consequences of analysing the untransformed scores; (5) heavy tails matter more than skew, so a transformation would need to address (and not make worse) any problems related to tail weight; and (6) typically distributions remain skewed after transformation and the more obvious transformations generally do not deal effectively with outliers (see Wilcox, 2017, for a review).

Adjusting standard errors

As we have seen, autocorrelation and heteroscedasticity lead to incorrect standard errors. Autocorrelation is easily dealt with by extending the model in Eq. 1 to one that explicitly models dependency in residuals (and, therefore, produces correct standard errors). For example, the model in Eq. 3 (which for simplicity contains only 1 predictor) includes terms that estimate the variance across contexts (e.g., time) in both the constant (ζ_{0i}) and model parameters for each predictor (e.g., ζ_{1i}). This model is an example of a multilevel model in which observations (*i*) are nested within contexts (*j*). These contexts could be individuals (e.g., repeated measures designs) or environments (e.g., classrooms). Autocorrelation between residuals is easily modelled, for example, by imposing a covariance structure that has sphericity in repeated measures designs, or specifying some other meaningful covariance structure such as a first-order autoregressive one).

$$\hat{Y}_{ij} = \hat{b}_{0i} + \hat{b}_1 X_{ij} + (\varepsilon_{ij} + \zeta_{0i} + \zeta_{1i} X_{ij}) \qquad \textit{Eq. 3}$$

There are also ways to adjust standard errors to be robust in the presence of heteroscedasticity. The standard OLS estimate of the standard errors of parameters uses the variance-covariance matrix of residuals (Φ). The assumption of independence implies that the covariances in Φ (the off-diagonal elements) are zero, and homoscedasticity implies that the variances (the diagonal elements) are equal to the variance in residuals (σ^2). Having estimated the model parameters ($\hat{b}$s), σ^2 can be estimated from the observed residuals and used to compute the standard errors for those parameters. When the variances are not equal, we cannot use σ^2 because the diagonal elements of Φ will differ. One solution is to use OLS to estimate the model and calculate the model residuals in the first instance, then to estimate Φ by placing the *i*th squared residual (e_i^2) into the *i*th row of the

diagonal of Φ and use this in the equation that is used to compute estimates of the standard errors (for a relatively non-technical explanation see Hayes & Cai, 2007). This method results in what are known as Eiker-White-Huber heteroscedasticity-consistent stranded errors (they are part of a family known as sandwich estimators). The resulting robust standard errors can be used to compute confidence intervals, test-statistics (and associated *p*-values) that are robust to heteroscedasticity.

Another way to deal with bias in standard errors and (confidence intervals) is to estimate them empirically. The Bootstrap (Efron, 1979, 1988; Efron & Tibshirani, 1993) is a flexible and general empirical method to find standard errors and confidence intervals for any statistic that is usually more accurate than traditional approaches. Bootstrapping works on a simple principle of estimating the shape of the sampling distribution by sampling with replacement from the data. The starting point is to create a bootstrap sample as large as the sample data by taking a value from the sample data and replacing it before sampling the next score. For example, imagine a sample of 5 scores on the SCARED anxiety scale for children (Birmaher et al., 1999): 8, 11, 15, 23, 29. The bootstrap sample will also have 5 scores, but it is created using sampling with replacement. So, the process might first randomly select the score 8 from the sample. Sampling with replacement means that, having been sampled, the value 8 is not taken from the pool of numbers that can be sampled but is available for selection when the next value is randomly selected. Therefore, you might end up with a bootstrap sample of 8, 8, 15, 23, 29. Note that the value 8 is sampled twice, and the value 11 has not been sampled at all. The process computes and stores the statistic of interest for that bootstrap sample (for example, the mean). Next a second bootstrap sample is created in the same way, perhaps it is 11, 11, 11, 15, 23 (note

that 11 has been sampled 3 times, and 8 and 29 have not been sampled at all). The statistic of interest is againcomputed and stored. This process is repeated until (usually) at least 1000 bootstrap samples have been created. It is commonly recommended that 2000 bootstrap samples are used. A computer can generate these bootstrap samples in seconds.

At the end of the bootstrapping process we are left with, say, 2000 bootstrap samples and from each we have an estimate of the statistic of interest (e.g., the mean). It is simple matter to estimate the standard error of the statistic by using the standard deviation of these bootstrap estimates. Similarly, the x% confidence interval for the statistic can be estimated from the values that enclose the middle x% of the bootstrap sample estimates. For example, the limits of the 95% bootstrap confidence interval for the statistic will be the values that enclose the middle 95% of the ordered bootstrap sample estimates. This is known as a percentile bootstrap confidence interval, but there are variants such as the BCa confidence interval, which corrects for skew. As with heteroscedasticity-consistent standard errors, bootstrap standard errors (and associated test statistics and *p*-values) and confidence intervals should be robust to violations of the assumptions we have discussed.

Alternative estimators

The assumptions that we have discussed relate to OLS estimation, therefore, one way to circumvent problems associated with this method is to use a form of estimation that does not assume homoscedastic and independent errors, or normality, or is robust to deviations from these assumptions. One family of robust estimators is based on the idea of trimming data, which involves removing extreme scores using a percentage of scores (the best-known example being the sample median).

It is quite common in experimental psychopathology research to do manual trims of the data based on outlier detection techniques (e.g., standard deviation based trims or idiosyncratic deletion). For example, it is the norm with reaction time data to use standard deviation based trims such as excluding scores greater than 2.5 standard deviations from the mean (Ratcliff, 1993). This approach is flawed because both the mean and standard deviation are highly influenced by outliers (whether overt ones, or covert ones such as in a mixed normal distribution). More precisely, this approach suffers from masking, meaning that outliers can be missed due to the sensitivity of the mean, and especially the standard deviation, to outliers. Manual inspection and removal of outliers is problematic too because it will typically be followed up by fitting a model to the remaining data that uses some OLS estimation method. Doing so results in an incorrect estimate of any standard errors. Regardless of how large the sample size happens to be, the resulting confidence intervals will be inaccurate.

With the understanding that no single method is always best, estimators based on percentage-based trims tend to perform well. For example, for a number of statistical methods a 20% trim (where the top and bottom 20% of scores are ignored) produce robust test statistics (Wilcox, 1998, 2010, 2017). Other variants of trimming include using the median (effectively trimming everything except the middle score), and *M-estimators*. M-estimators determine whether a score is an outlier empirically and if it is, adjustments are made for it. The adjustment could be to completely ignore the observation or to down-weight it. Obvious advantages of M-estimators are that you can (1) down-weight rather than exclude observations; (2) avoid over- or under-trimming your data; and (3) perform non-symmetric trimming (although this issue is not straightforward). For a more technical

discussion of M-estimators see Wilcox (2017). As with the other trimming methods, you cannot simply trim the data and apply OLS estimation, but appropriate methods for M-estimators and percentage trims are available (unlike for *SD* based and idiosyncratic trims).

We have already mentioned that when the assumptions of independent, homoscedastic and normally-distributed errors are met the OLS estimator will also be the maximum likelihood estimator. When these assumptions are not met, the ML estimator will yield different results to the OLS. The ML estimator is a lot more versatile than OLS and tends to be the default for more complex variants of the linear model (such as multilevel models, models with latent variables etc.). Given that any variant of the linear model can be expressed in a structural equation modelling framework, ML estimation could be used to estimate the vast majority of research designs in experimental psychopathology. ML is robust to small departures from normality but (as with OLS) standard errors and significance tests can be adversely affected by more severe departures from normality (especially kurtosis), and heterogeneity (see Brown, 2015, for a review). In such situations, robust variants of ML estimation can be used that adjust standard errors using the Huber-White method already described and scale the test statistic (MLR estimation). The MLR estimator is also robust to violations of the assumption of independence. There are other variants (such as MLM, MLMVS and MLMV) but these can be used only with complete data and so we will focus on MLR.

Weighted least squares (WLS) estimation and its robust variant (WLSMV) is also useful when homoscedasticity cannot be assumed. The idea with WLS is that observations are weighted by some function of their precision thus allowing more precise observations to contribute more to the parameter estimates. However, this method assumes that weights

are known exactly (which they almost never are and instead they must be estimated). WLS is also highly sensitive to outliers and performs poorly in small-to-medium sized samples (Brown, 2015).

Examples using R

In the remainder of this paper we will present worked examples relevant to clinical psychology and experimental psychopathology researchers using the software R (R Core Team, 2016). We present the results from OLS models and a robust counterpart (but not *the only* robust counterpart). We make this comparison only to frame the models in familiar territory. All but one of the data sets are from published studies and the assumptions we have discussed were not substantially violated. As such, the conclusions from the robust models would not be expected to conflict with those from the classical models, but act as a useful sensitivity analysis.

Using R

There is not the space either to describe the basics of how to use R, or cover all of the more than 1200 functions in the R package described in Wilcox (2017), but we will give you a flavour using a tiny selection[iii]. You can find a lot more depth and breadth about R and the functions for robust statistical analysis in our books (Field et al., 2012; Wilcox, 2016, 2017) and others. A few important things to know about R: (1) it has base functionality that is extended by installing (using the function **install.packages()**) and loading (using the function **library()**) packages; (2) it is case sensitive so when faced with errors check that letters that need to be capitalized have been and vice versa; and (3) variables collected together into an object (like a spreadsheet) are known as a dataframe.

The data files, R scripts and other materials accompanying this paper are posted at https://osf.io/fbj3z/.

Initialising the packages

We will use the WRS2 (Mair, Schoenbrodt, & Wilcox, 2017) and robustbase (Rousseeuw et al., 2015) packages to access functions for some robust tests, lme4 (Bates, Maechler, Bolker, & Walker, 2015) and robustlmm (Manuel Koller, 2016) for the multilevel model, and lavaan (Rosseel, 2012) for the latent growth model. To access these packages, execute the following (the 'install.packages' commands are necessary only if you do not already have the packages installed)[iv]:

```
install.packages("lavaan");    install.packages("lme4");    install.packages("robustlmm");
install.packages("WRS2")

library(lavaan)
library(lme4)
library(robustbase)
library(robustlmm)
library(WRS2)
```

Loading datasets into R

The examples in this tutorial use 4 datasets described below that are saved as CSV files. To load each dataset into R, do one of two things. Option 1 is to adapt and execute this generic command:

```
name<-read.csv(file.choose())
```

In which you replace 'name' with a word that you wish to use to name the dataframe containing the contents of the CSV file. The function **file.choose()** opens a standard dialogue box that enables you to navigate to the CSV file that you want to load. Option 2 is to place the data files in a single folder, set the working directory to be that folder and adapt and execute:

```
name<-read.csv("nameOfFile.csv")
```

In which you replace ‘name’ with a word that you wish to use to name the dataframe, and replace ‘*nameOfFile*’ with the name of the CSV file that you wish to load. For example, to import the file *FieldLawson2003.csv* into a dataframe called ‘*fieldWide*’, execute either of these commands:

```
fieldWide <-read.csv(file.choose())
fieldWide <-read.csv("FieldLawson2003.csv")
```

Both create a dataframe called ‘*fieldWide*’ containing the data in the CSV file but the first does this through you navigating to the file *FieldLawson2003.csv* and the second does it by looking for that file in the working directory.

Descriptions of the datasets

Fear learning in children (Field & Lawson, 2003)

The file *FieldLawson2003.csv* contains the data from the behavioural avoidance task of Field and Lawson (2003), published in this journal. In this experiment, children aged 6 to 9 years were given verbal information about two novel Australian marsupials that contained either threat or positive content. A third marsupial, about which no information was given, acted as a control. (The type of information was counterbalanced across animals for different children). After the information, children were asked to approach three boxes that they were told contained the animals (in fact they did not). Latency to approach the boxes acted as a behaviour measure of their fear of these animals. This part of the experiment has a one-way repeated measures design (children approached all three boxes). The approach times were reported as *z*-scores where a positive score indicates that children took longer than average to approach, 0 represents the average approach time, and a negative score is indicative of being faster than average to approach. These data are shown

in Figure 3 of the original paper (split by the biological sex of the child, which we will ignore).

The dataset contains 4 variables: 'id' indicates the participant code and 'zThreat', 'zPos' and 'zNone' are the *z*-scored approach times for each child to approach the box containing the animal about which they were given threat, positive or no information respectively. A version of this data file in 'long' format (*FieldLawson2003Long.csv*) contains these data restructured into three variables: (1) 'id' as above; (2) 'InfoType' codes whether a score relates to an animal about which threat, positive or no information was given, and (3) 'value' contains the z-score for the time for a given child to approach a given box. Load these files into dataframes called '*fieldWide*' and '*fieldLong*' using the instructions above.

Fear unlearning in children (Kelly, Barker, Field, Wilson, & Reynolds, 2010)

The file *Kellyetalz.csv* contains the data from a paper also published in this journal by (Kelly et al., 2010) that investigated whether verbal information or modelling were effective in reversing the effect of verbal threat information on children's fears of novel animals. Like the previous experiment, children aged 6 to 8 years old were given threat information or no information about two novel Australian marsupials. Following this information, different groups received one of three 'interventions': (1) positive information about the threat animal; (2) a positive modelling experience (an adult placing their hand in a box seemingly containing the threat animal); or (3) no further experience. The children's 'fear' of the marsupials was measures using a self-report measure called the Fear Beliefs Questionnaire (FBQ) or a behavioural approach task (BAT) like that described above. In

the paper, the authors test the specificity of the interventions by comparing their effects on the subjective (FBQ) and behavioural (BAT) components of the fear emotion. To do so, a single score was computed separately for the FBQ and BAT that represented the change from pre- to post-intervention for the threat animal relative to the control animal. These scores, therefore, represent the overall effect of the intervention on each measure. The scores were converted to *z*-scores separately for the FBQ and BAT so that they could be compared. The means of these scores are displayed in Figure 3 of the original paper. Note that because the interventions are expected to reduce fear, greater efficacy is shown up by more *negative z*-scores (i.e. greater *reductions* in fear). This part of the study had a mixed design with a between group manipulation of intervention (positive information, non-anxious modelling, no intervention) and a repeated measures manipulation of the type of measure (FBQ or BAT).

The data file contains 4 variables: (1) 'id' indicates the participant number; (2) 'Intervention' is a factor indicating whether the child received positive information, non-anxious modelling or no intervention, (3) 'Measure' indicates whether a score came from the FBQ or BAT; and (4) 'z' is the z-score associated with the measure. Note that FBQ and BAT scores are in a single column rather than two columns, which is known as the long format (contrasted with the wide format with which SPSS users will be more familiar). Load the data into a dataframe called '*kellyz*' by adapting the instructions above.

Predictors of social anxiety (Field & Cartwright-Hatton, 2008)

The file *FieldCH2008.csv* contains the data from a paper that looked at cognitive components of anxiety symptoms as predictors of social anxiety in 559 individuals. We used structural equation modelling to look at whether cognitive measures associated with

different anxiety disorders (e.g., worry in generalized anxiety, obsessive beliefs in obsessive compulsive disorder) also predicted social anxiety and whether they did so as unique predictors or via a latent variable representing general 'iterative thinking'. In the current context, we will use only the scale totals and look at a linear model that predicts social anxiety from four predictors.

The data file contains 6 variables: (1) 'id' contains the participant code; (2) 'socAnx' contains social anxiety scores measured by the SPAI (Turner, Beidel, & Dancu, 1996), (3) 'worry' contains worry scores measured by the PSWQ (Meyer, Miller, Metzger, & Borkovec, 1990), (4) 'shame' contains scores measuring shame measured by the TOSCA-3 (Tangney, Dearing, Wagner, & Gramzow, 2000), (5) 'imagery' contains scores from a measure of visual imagery, the VVIQ (Marks, 1973), and (6) 'obsessive' contains scores measuring the participant's obsessive beliefs using the OBQ (Steketee et al., 2001). These are measures commonly used by researchers publishing in this journal. Load the data into a dataframe called '*fieldCH*' by adapting the instructions above.

Generic RCT design

The file *RCTWide.csv* contains data that mimics a generic RCT design (N = 200) in which two randomized groups (cognitive behaviour therapy, CBT, and treatment as usual, TAU) have measures of a mental health outcome (scored 0-100) taken before a treatment (baseline), after a 2-month CBT intervention (or TAU) and again 6 months after treatment had ended (8 months post-baseline).

The data file contains 6 variables: (1) 'ID' contains participant ids; (2) 'Group' identifies whether the client was randomized to CBT or TAU, (3) 'Baseline' contains outcome scores pre-treatment, (4) 'FU_2_Month' contains outcome scores post-treatment,

and (5) ‘FU_8_Month’ contains outcome scores 6 months post-treatment (8 months post baseline). A version of this data file in ‘long’ format (*RCTLong.csv*) contains these data restructured into four variables: (1) ‘ID’ as above; (2) ‘Group’ as above, (3) ‘Time’ indicates the time (in months) at which an observation was taken (0, 2 or 8), and (4) ‘Outcome’ contains the observed outcome scores. Load these files into dataframes called ‘*rctWide’* and ‘*rctLong*’ respectively by adapting the instructions above.

Viewing the data

To view the data in the dataframes you have just created, execute the dataframe’s name, or use the **head()** function to look at some cases at the top of the dataframe. For example, executing the first command below will print the entire *fieldWide* dataframe to the screen and executing the second will print the first 10 cases (rows) in the *fieldWide* dataframe:

```
fieldWide
head(fieldWide, 10)
```

Example 1: Comparing two dependent means

First we will look at how to compare two dependent means using the **yuend()** function, which implements Yuen’s modified *t*-test for trimmed means (Yuen, 1974). The function takes the general form:

```
yuend(dataForCondition1, dataForCondition2, tr = 0.2)
```

In which the option ‘tr’ specifies the level of trimming as a proportion. The default of ‘tr = 0.2’ will compare 20% trimmed means, which is recommended (Wilcox, 2017). The other parts of the function specify the variables containing the two conditions that we wish to compare. For example, in the Field and Lawson (2003) data, imagine that we wanted to compare the mean latency to approach the threat animal against the mean latency

to approach the control animal. We use the *fieldWide* dataframe that you have already created. The data for the threat animal are stored in *fieldWide$zThreat*, and in *fieldWide$zNone* for the control animal[v], so if we leave the default level of 20% trimming (by excluding this option) we can run the test by executing:

```
yuend(fieldWide$zThreat, fieldWide$zNone)
```

We can compare this test with the regular dependent *t*-test, by executing:

```
t.test(fieldWide$zThreat, fieldWide$zNone, paired = T)
```

Figure 1 shows that both tests yield significant differences. Note that trimming reduces the mean difference from 0.49 to 0.40, and that the test statistic is smaller in the robust version. We could report the robust test as a significant difference between trimmed mean approach times to the threat and control animals, M_{diff} = 0.40 [0.08, 0.74], Y_t (26) = 2.53, p = 0.02.

Example 2: comparing several dependent means

We can compare latencies to all three boxes (i.e., compare three dependent means) using the **rmanovab()** function and get post hoc tests with **pairdepb()**. These functions require the data in long format (latencies for different boxes in a single column). The data that you loaded into the *fieldLong* dataframe are in this format. Both **rmanovab()** and **pairdepb()** take a similar form:

```
rmanovab(outcomeVariable, conditionVariable, idVariable, tr = 0.2, nboot = 599)
```

As before, the option '*tr*' controls the amount of trim (and the default of 20% is advised), but because these functions use a bootstrap there is an additional option, *nboot*, to control the number of bootstrap samples. The default of 599 is sufficient, but it is common to use 1000 or 2000. For the current data, the outcome variable is *fieldLong$value*, the variable that codes to which box each latency score relates is *fieldLong$InfoType*, and

the variable that identifies the participant within which scores are nested is *fieldLong$id*. Therefore, if we use a 20% trim (by omitting this option) but increase the number of bootstrap samples we would execute:

```
rmanovab(fieldLong$value, fieldLong$InfoType, fieldLong$id, nboot = 2000)
pairdepb(fieldLong$value, fieldLong$InfoType, fieldLong$id, nboot = 2000)
```

We can compare this method to the conventional one-way repeated measures ANOVA with Type III sums of squares, which is obtained by executing[vi]:

```
summary(aov(value ~ InfoType + Error(id/InfoType), data = fieldLong))
pairwise.t.test(fieldLong$value, fieldLong$InfoType, p.adjust.method = "bonferroni", paired
= T)
```

In Figure 2 we can see from the conventional linear model (ANOVA) that the means are significantly different with latencies after threat information being significantly longer than for positive or no information. The robust test has a test statistic, critical value of the test statistic (at $\alpha = .05$) and whether the test is significant at the .05 level (i.e. does the observed statistic exceed the critical value). We would report that there was a significant difference between trimmed mean approach times to the three animals, $F_t = 6.75$, $F_{Crit} = 3.13$, $p < .05$. The post hoc tests tell us the difference between trimmed means (*psihat*), and the associated bootstrap confidence interval, the test of this difference, the critical value of the test and whether the trimmed means are significantly different (at $\alpha = .05$). We would report that the trimmed mean difference in latency between the threat box and the positive, $\hat{\psi} = 0.52$ [0.15, 0.90], and no information, $\hat{\psi} = 0.40$ [0.01, 0.79] boxes were significant. The trimmed mean difference between the positive and the no information box was not, $\hat{\psi} = -0.12$ [–0.44, 0.20].

Example 3: comparing two independent means

We now turn to the Kelly et al. data. First, we will look at how to do a robust comparison of two means using the **yuenbt()** function, which implements Yuen's modified *t*-test for independent trimmed means (Yuen, 1974) with a bootstrap. We are going to compare the FBQ *z*-scores in two conditions: positive information vs. No intervention. To do this, we need to first select this subset of the data by executing these commands:

```
posInfoFBQ<-subset(kellyz, Intervention != "Non-Anxious Modelling" & Measure == "FBQ")
posInfoFBQ$Intervention<-factor(posInfoFBQ$Intervention)
```

The first command creates a new dataframe called *posInfoFBQ* using the **subset()** function. Within this function, we take the *kellyz* dataframe and select cases based on the logical condition that the variable 'Intervention' is not equal to (!=) the value 'Non-Anxious Modelling' and the variable 'Measure' is equal to (==)[vii] the value 'FBQ'. The resulting dataframe will match the *kellyz* dataframe except that children in the intervention group that had a positive modelling experience will have been removed, as will scores relating to the BAT. By executing the name of the dataframe you will see that 'Intervention' now contains two groups (not three) and the variable 'Measure' contains only 'FBQ'. One problem is that the variable 'Intervention' will still be coded as having 3 levels (R will treat the 'modelling' condition as if it contains no data). The second command re-sets the coding for the variable 'Intervention' by re-creating it from itself as a factor. 'Intervention' will now have two levels corresponding to the groups containing data (positive information and no intervention).

We conduct the robust test using the **yuenbt()** function, which takes the general form:

```
yuenbt(outcomeVariable ~ groupVariable, data = dataframeName, tr = 0.2, nboot = 599)
```

The options '*tr*' and '*nboot*' have already been described and we will assume from heron that you know what they do and how to use them. All we need to do then, is specify the outcome variable name (in this case '*z*'), the name of the variable representing group membership (in this case '*Intervention*'), and the name of the dataframe:

```
yuenbt(z ~ Intervention, data = posInfoFBQ, nboot = 2000)
```

We can compare this test with a classic *t*-test for independent means, which is obtained by executing:

```
t.test(z ~ Intervention, data = posInfoFBQ)
```

Figure 3 shows that both tests yield highly significant results. We could report the robust test as a significant difference between trimmed mean FBQ *z*-scores in the positive information intervention compared to no intervention, M_{diff} = 1.39 [0.91, 1.88], Y_t = 6.07, p = 0.00.

Example 4: comparing several independent means

The previous analysis was illustrative: our hypothesis about threat information would require a comparison of trimmed FBQ means across all three intervention groups (in ANOVA terminology this linear model would be labelled a one-way independent ANOVA). This model is fit using the **t1waybt()** function and the **mcppb20()** function to get post hoc tests. As in the previous example, we create a new dataframe that includes only the FBQ data (i.e. removes the BAT scores). We achieve this in a similar way to the previous example by executing:

```
fbqOnly<-subset(kellyz, Measure == "FBQ")
```

This command creates a new dataframe called '*fbqOnly*' using the **subset()** function to take the *kellyz* dataframe and select cases if the variable 'Measure' is equal to the value

'FBQ'. The resulting dataframe will match *kellyz* except that the BAT scores have been removed.

The functions **t1waybt ()** and **mcppb20 ()** take a similar form to **yuenbt()**:

```
t1waybt(outcomeVariable ~ groupVariable, data = dataframeName, tr = 0.2, nboot = 599)
```

For the current data, the outcome variable is 'z', the variable that codes which intervention a child received is 'Intervention' and the dataframe is called 'fbqOnly'. If we accept the default of a 20% trim (by omitting this option) and request 2000 bootstrap samples we would execute:

```
t1waybt(z ~ Intervention, data = fbqOnly, nboot = 2000)
mcppb20(z ~ Intervention,  data = fbqOnly, nboot = 2000)
```

We will compare the robust test to the classic linear model, which can be obtained by executing:

```
summary(aov(z ~ Intervention, data = fbqOnly))
pairwise.t.test(fbqOnly$z, fbqOnly$Intervention, p.adjust.method = "bonferroni")
```

Figure 4 shows that the type of information significantly predicted FBQ scores in both models, with FBQ scores after all types of information differing significantly from each other. The robust test produces an effect size (essentially a robust analogue of Pearson's correlation, Wilcox, 2017). We would report that there was a significant difference between the trimmed mean FBQ scores from the intervention groups, $F_t = 19.90$, $p < .001$. The post hoc tests tell us the difference between trimmed means, the associated bootstrap confidence interval, and the *p*-value for this difference. We would report that (based on the trimmed mean difference in FBQ scores) the intervention was significantly more effective for positive information than modelling, $\hat{\psi} = 0.38$ [0.07, 0.76], and no intervention, $\hat{\psi} = 1.39$ [0.87, 1.89], and for modelling compared to no intervention, $\hat{\psi} = 1.01$ [0.42, 1.57].

Example 5: a two-way mixed design

In the published paper, Kelly et al. compared the effects of the three interventions on behavioural (BAT) and subjective (FBQ) components of fear using a 2 × 3 mixed design. In the paper itself we used a method based on trimmed means and a bootstrap, but for simplicity here we will demonstrate the **bwtrim()** function, which does not apply a bootstrap process. The function is like ones we have seen previously:

```
bwtrim(outcomeVariable ~ betweenVariable*withinVariable, id = idVariable, data = dataframeName, tr = 0.2)
```

The outcome variable is '*z*', the variable representing the between-group variable is called 'Intervention', and the repeated-measures variable is called 'Measure'. We need to specify the variable that represents the case to which a score belongs (i.e., a variable that identifies the participant), which is the variable called 'id'. Finally, we specify the name of the dataframe (*kellyz*). If we use the default 20% trim then we would execute:

```
bwtrim(z ~ Intervention*Measure, id = id, data = kellyz)
```

The classical linear model for this design (a 2 × 3 mixed ANOVA to use a common label) is obtained by executing[viii]:

```
summary(aov(z ~ Intervention*Measure + Error(id/Measure), data = kellyz))
```

Figure 5 shows the details of the models. Like in Kelley et al.'s paper, the robust test yields a significant main effect of the intervention group ($p < .001$), and the interaction between the intervention condition and the measure used ($p = .003$), but not of the main effect of measure ($p = .108$). This mirrors what is found using a classical test (which is not surprising given how significant the effects are).

Example 6: robust linear models with continuous predictors (regression)

Although not the analysis performed in the original paper, we will use the data from Field and Cartwright-Hatton (2008) to look at the extent to which social anxiety can be

predicted from measures of worry, shame, visual imagery and obsessive beliefs. This example is particularly important because it demonstrates a robust linear model with multiple predictors (i.e., multiple regression), and a great many variants of research designs can be conceptualised as linear models in which groups are coded using dummy variables (see Field, 2013, 2016; Field et al., 2012, or many other introductory books). As we will see, this example provides a foundation for robust variants of independent ANOVAs and ANCOVAs). There are many methods that deal with heteroscedasticity, outliers and even curvature (the assumption of linearity is inaccurate).

Let us first run the OLS model. To do this we use the **lm()** function in R, which at its most basic, takes the form:

```
lm(outcomeVariable ~ predictorVariable1 + predictorVariable2 + … predictorVariableN, data = dataframeName)
```

The function's input is, essentially, the equation describing the linear model that you want to fit. That is, you specify the variable containing the outcome scores on the left of the tilde, then on the right of it list the names of any predictor variables from the dataframe with each one separated by a +. We have used this formula input several times in previous examples. You also specify the name of the dataframe from which these variables come (*fieldCH* in this case). To predict the variable 'socAnx' from the variables 'worry', 'shame', 'imagery' and 'obsessive' we would, therefore, execute:

```
socAnx.normal<-lm(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
summary(socAnx.normal)
```

Note that rather than executing the function directly, we use it to create an object (which I have called *socAnx.normal*). The first line creates this object from the **lm()** function, and in doing so 'socAnx.normal' contains the model parameters and other useful information. We can access a summary of the model by executing the second command,

which asks R to display a summary of the object ‘socAnx.normal’. It is possible to wrap these two lines into a single command as we have in previous examples by inserting line 1 into the parentheses of line 2. The resulting summary is shown in the top of Figure 6.

There are several options for fitting robust regression in R. We will demonstrate the **lmrob()** function, which fits a robust variant of the social anxiety model based on an M-estimator (M. Koller & Stahel, 2011; Yohai, 1987) using iteratively reweighted least squares (IRWLS) estimation. This function, at its most basic, takes the same form as **lm()**, which means that we can simply replace ‘lm’ with ‘lmrob’ and proceed as before.

```
socAnx.robust<-lmrob(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
summary(socAnx.robust)
```

Compare these commands to those for the non-robust regression and you can see that apart from changing the model name to *socAnx.robust*, all that is different is that ‘lm’ has changed to ‘lmrob’ – it is that simple.

Comparing the model parameters for the OLS and robust models (Figure 6) note that the *b*-values (in the column labelled ‘Estimates’), standard errors, *t*-values and *p*-values are slightly different. The interpretation of the model does not change substantially (worry, visual imagery and obsessive beliefs significantly predict social anxiety, shame does not) but the parameter estimates and associated standard error, test statistic and *p*-value from the robust model will have been relatively unaffected by the shape of the model residuals and outliers etc.

Given that our earlier examples are also variants of the linear model, we could also use the *lmrob()* function is we wanted to use an *M*-estimator instead of trimmed means. For example, the classical models that compared two independent means (example 3) and several independent means (example 4) were obtained using the **aov()** function, but this

function is a wrapper for the **lm()** function that expresses the model in terms of *F*-statistics (as in ANOVA) rather than model parameters. If we use the **lm()** function directly to fit these models we obtain the model parameters in Figure 7:

```
summary(lm(z ~ Intervention, data = posInfoFBQ))
summary(lm(z ~ Intervention, data = fbqOnly))
```

It is a simple matter to estimate these parameters with an M-estimator by replacing 'lm' with 'lmrob':

```
summary(lmrob(z ~ Intervention, data = posInfoFBQ))
summary(lmrob(z ~ Intervention, data = fbqOnly))
```

The output in Figure 7 shows the robust and non-robust variants of both models. Given the highly significant effect of the type of information on children's approach times it is unsurprising that the conclusions from the robust models mimic those from the classical ones. However, note the differences in the model parameters and their standard errors, which will be more accurate in the robust models under violations of test assumptions.

Example 7: robust multilevel models

The final dataset concerns a RCT design (see earlier). This design could be conceptualised in various ways. For example, you could fit a linear model with Group (RCT vs TAU) and Time (baseline, 2 months and 8 months) as independent variables with repeated measures on the time variable (see example 5). However, this is a special case of a multilevel model in which observations (level 1) are nested within clients (level 2) where a spherical covariance structure is assumed. When observations are not spaced evenly over time (and indeed in other situations) we might want to relax this assumption or model other forms of covariance structure such as a first-order autoregressive structure. We can express this model by extending Eq. 3 to Eq. 4.

$$\hat{Y}_{ij} = [\gamma_{00} + \gamma_{10}\text{Time}_{ij} + \gamma_{01}\text{Group}_i + \gamma_{11}(\text{Group}_i \times \text{Time}_{ij})] + [\zeta_{0i} + \zeta_{1i}\text{Time}_{ij} + \epsilon_{ij}] \quad \textit{Eq. 4}$$

The structural part of the model (first set of square brackets) states that the outcome in participant *i* at time *j* is predicted from the intercept plus the rate of change for that participant *i* at time *j*, group membership of the participant, and the interaction of group membership and the rate of change at time for participant *i* at time *j*. The stochastic part (second set of square brackets) includes terms representing the difference between the individual's intercept and that of the population average (ζ_{0i}), the difference between the individual's rate of change (slope) and that of the population average (ζ_{1i}), and a term allowing for random scatter of the individual's data around their particular trajectory (ϵ_{ij}). We can create a classical model (*rctLmer*) using the **lmer()** function and a robust variant (*rctRLmer*) using **rlmer()**. In both cases the outcome is predicted from the fixed main effects of Group and Time and their interaction (*Outcome~Group*Time*), we also allow the intercept and slope of time to vary by participant *(+ (Time|ID))*; in doing so we are allowing participants to have different growth trajectories. To get ML estimation we set *REML = FALSE*.

```
rctLmer<-lmer(Outcome~Group*Time + (Time|ID), data = rctLong, REML = FALSE)
summary(rctLmer)

rctRLmer<-rlmer(Outcome~Group*Time + (Time|ID), data = rctLong, REML = FALSE)
summary(rctRLmer)
```

Figure 8 shows the outputs from the two models. Note that the parameter estimates differ because of the robust estimation used in the later model (for example, γ_{01} the parameter for the main effect of group is 2.35 in the classical model but 2.49 in the robust one).

Example 8: robust latent growth models

The model in the previous example can also be expressed as the latent growth model in Figure 9. Essentially, the slope and intercept for the growth trajectory (the change in the outcome over time) are conceptualised as latent variables that are estimated from the observes outcome at different time points, and can be influenced by the group to which a person belonged. To specify this model using the *lavaan* package in R, we first need to specify the model in Figure 9 as:

```
rctModel <- '
i =~ 1*Baseline + 1*FU_2_Month + 1*FU_8_Month
s =~ 0*Baseline + 2*FU_2_Month + 8*FU_8_Month

#variances/covariances
i ~~ i
s ~~ s
i ~~ s

#intercepts
i ~ 1
s ~ 1
Baseline ~ 0
FU_2_Month ~ 0
FU_8_Month ~ 0

#Predictors
i + s ~ Group
'
```

This command creates a new object called *rctModel*, which contains the model specification in single quotes. The first two lines define the intercept (*i*) and slope (*s*) as being indicated from the observed outcome variables at baseline (*Baseline*), after therapy (*FU_2_Month*) and 6 months after therapy (*FU_8_Month*). Note that for the slope these variables are weighted by the months from baseline (0, 2 and 8). The next section defines the variances for the intercept (*i ~~ i*) and slope (*s ~~ s*) and their covariance (*i ~~ s*). The

predictors section specifies that the intercept and slope are predicted from the observed variable Group (*i + s ~ Group*).

Having specified the model we can feed the object *rctModel* into the **growth()** function. By default, this function used ML estimation:

```
rctFit <- growth(rctModel, data = rctWide)
summary(rctFit)
```

To change Ml to one of the robust estimation methods discussed earlier, such as MLR, MLM, MLMVS, MLMV, WLS and WLSMV, we add the *estimator* command. For example, to fit the model using MLR estimation we could execute (note that all that has changed is that we have added a command to override the default ML estimator):

```
rctMLR <- growth(rctModel, data=rctWide, estimator = "MLR")
summary(rctMLR)
```

For WLSMV estimation we could execute (although the sample is too small for these data):

```
rctMLMVS <- growth(rctModel, data=rctWide, estimator = "MLMVS")
summary(rctMLMVS)
```

We could also bootstrap confidence intervals and standard errors by executing:

```
rctBoot <- growth(rctModel, data=rctWide, se = "bootstrap")
summary(rctBoot)
```

The point is that any model that you can specify as a path model, you can fit with robust variants of ML estimation.

Figures 9 and 10 show the outputs using the ML estimator and MLR estimator respectively. Figure 9 highlights the key substantive effects of the main effect of time, and the effect of group on both the slope and intercept of the growth curve over time. In particular, the effect of group on the slope indicates whether the change in the outcome over time was different in the CBT and TAU groups. If CBT has had an effect we would

expect a steeper (negative) slope in the CBT group compared to the TAU (indicating a sharper decline in symptoms for the CBT group). By comparing Figures 9 and 10 you can see how using a robust estimator affects the parameter estimates, their standard errors and significance tests. Given the highly significant effects the choice of estimator does not ultimately yield different conclusions based on the significance tests, but does yield more robust parameter estimates (which is helpful for future prediction).

Discussion and recommendations

The four aims of this paper are to convince you that (1) the assumptions of the statistical model that is very frequently used in papers published in this journal are highly unlikely to be met for psychological data; (2) violating these assumptions has unpleasant and undesirable effects on the model parameters and their associated standard errors, confidence intervals and *p*-values; (3) traditional methods for dealing with violations of model assumptions are ineffective; and (4) there are numerous robust alternatives to the models that researchers in this area typically use and they are straightforward to implement. Briefly, the possible practical consequences of violating assumptions include relatively low power, inaccurate confidence intervals and measures of effect size that miss important differences.

Given our conviction that violations of assumptions are the norm rather than the exception it is counter-intuitive that the status quo when reporting data is not to report on the likelihood of assumptions having been met and in the absence of this information assuming that they have. Given Micceri's (1989) findings it seems highly improbable that every paper not explicitly demonstrating that model assumptions have been met have, in reality, met the model assumptions. Our first recommendation is, therefore, that reviewers

and editors reverse the current assumption and instead assume that assumptions have *not* been met unless there is an explicit and compelling statement, backed up by evidence, that the assumptions of the models fit have been met.

On what should such statements be based? First, we do not recommend that these statements are based upon significance tests of assumptions because, under general conditions, such tests do not have enough power to detect violations of assumptions that have practical consequences (Kesleman, Othman, & Wilcox, 2016). Currently the best way of investigating the impact of violations of model assumptions is to use a modern robust method and compare the results to the standard model. If the assumptions are met, the expectation is that they will give consistent results. Otherwise, the conventional method is in doubt.

We would also recommend against arguments for conventional methods based on sample-sized based arguments because, as we have explained, large sample sizes do not necessarily save the day. Also, it is not straightforward when large sample sizes suffice when using conventional methods. An additional concern is that when distributions differ in other ways, such as skewness, poor control over the Type I error probability can occur even with large sample sizes. Imagine a two-sample Student's *T*, with a sample size of 400 in the first groups from a lognormal distribution, and sample size of 1000 in the first group from a normal. Both groups have the same mean and variance equal to 1. The Type I error probability is approximately .14 rather than the nominal .05; this has been known for more than half a century (Pratt, 1964). Regardless of how large the sample size might be, results can be misleading.

Given these points it is hard to imagine what a compelling statement that test assumptions have been might look like, especially when you consider the impact of heavy tails on test power (and so on), and the complexity of ascertaining the precise impact that the specific degree of skew and kurtosis that a researcher faces might have.

Our second recommendation is, consistent with the default belief that test assumptions will not have been met, that (other things being equal) reviewers and editors insist on sensitivity analysis for all frequentist analyses. In other words, models based on non-robust estimators (such as OLS and ML) are compared to a robust variant. Where the two models yield ostensibly the same results then either model may be reported, where the models deviate substantially then the robust model should be reported unless a compelling evidence-based case can be made that model assumptions have been met. The details of such a sensitivity analysis could be made available on a public open science repository such as the OSF (https://osf.io/) for both reviewers and end users.

One objection to this recommendation might be that situations could arise in which no robust procedure is available to test the substantive hypotheses of the research. As highlighted in this article, there are over 1200 procedures available, covering the vast majority of research designs used by scientists submitting to this journal. Even when a specific test is unavailable, it should be technically possible to bootstrap standard errors and confidence intervals from pretty much any model.

Our final recommendation is that statements along the lines of 'ANOVA/regression/the *t*-test is robust' should be banned because based on hundreds of papers published during the past fifty years, it is well established that this statement is not correct (see Wilcox, 2017, for a review). Establishing blanket robustness is extremely hard

because of the variety of ways in which model assumptions can be violated. The safest option is always to consider results based on robust methods. In general, robust methods, including a range of techniques not covered here, provide the opportunity to get a deeper, more accurate and more nuanced understanding of data. At the moment, the only known method for judging the extent conventional methods yield reasonable results is to determine whether results are consistent with those based on robust methods. If authors justify applying the GLM by asserting its robustness, then they should evidence this belief through comparisons to an equivalent robust model.

References

Arostegui, I., Nunez-Anton, V., & Quintana, J. M. (2007). Analysis of the short form-36 (SF-36): The beta-binomial distribution approach. *Statistics in Medicine, 26*(6), 1318-1342. doi:10.1002/sim.2612

Barber, J. A., & Thompson, S. G. (2000). Analysis of cost data in randomized trials: an application of the non-parametric bootstrap. *Statistics in Medicine, 19*(23), 3219-3236.

Bates, D., Maechler, M., Bolker, B. M., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1-48. doi:10.18637/jss.v067.i0

Binks, C. A., Fenton, M., McCarthy, L., Lee, T., Adams, C. E., & Duggan, C. (2006). Psychological therapies for people with borderline personality disorder. *Cochrane Database of Systematic Reviews*(1). doi:Cd00565210.1002/14651858.cd005652

Birmaher, B., Brent, D. A., Chiappetta, L., Bridge, J., Monga, S., & Baugher, M. (1999). Psychometric properties of the Screen for Child Anxiety Related Emotional Disorders (SCARED): A replication study. *Journal of the American Academy of Child and Adolescent Psychiatry, 38*(10), 1230-1236. doi:10.1097/00004583-199910000-00011

Brown, T. A. (2015). *Confirmatory factor analysis for apploied research* (2nd ed.). New York: Guilford.

Canty, A., & Ripley, B. (2016). boot: Bootstrap R (S-Plus) Functions. R package version (Version 1.3-18). Retrieved from http://cran.r-project.org/package=boot

Cohen, J. (1968). Multiple regression as a general data-analytic system. *Psychological Bulletin, 70*(6), 426-443.

Donaldson, T. S. (1968). Robustness of the *F*-test to errors of both kinds and the correlation between the numerator and denominator of the *F*-ratio. *Journal of the American Statistical Association, 63*, 660-676.

Efron, B. (1979). Bootstrap methods - another look at the jackknife. *Annals of Statistics, 7*(1), 1-26. doi:10.1214/aos/1176344552

Efron, B. (1988). Bootstrap confidence-intervals - good or bad. *Psychological Bulletin, 104*(2), 293-296.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*: Chapman and Hall.

Field, A. P. (2013). *Discovering statistics using IBM SPSS Statistics: And sex and drugs and rock 'n' roll* (4th ed.). London: Sage.

Field, A. P. (2016). *An adventure in statistics: the reality enigma*. London: Sage.

Field, A. P., & Cartwright-Hatton, S. (2008). Shared and Unique Cognitive Factors in Social Anxiety. *International Journal of Cognitive Therapy, 1*(3), 206-222. doi:10.1680/ijct.2008.1.3.206

Field, A. P., & Lawson, J. (2003). Fear information and the development of fears during childhood: effects on implicit fear responses and behavioural avoidance. *Behaviour Research and Therapy, 41*(11), 1277-1293. doi:10.1016/s0005-7967(03)00034-2

Field, A. P., Miles, J. N. V., & Field, Z. C. (2012). *Discovering statistics using R: And sex and drugs and rock 'n' roll*. London: Sage.

Glass, G. V., Peckham, P. D., & Sanders, J. R. (1972). Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance. *Review of Educational Research, 42*(3), 237-288.

Grayson, D. (2004). Some myths and legends in quantitative psychology. *Understanding Statistics, 3*(1), 101-134.

Hayes, A. F., & Cai, L. (2007). Using heteroskedasticity-consistent standard error estimators in OLS regression: An introduction and software implementation. *Behavior Research Methods, 39*(4), 709-722.

Hlatky, M. A., Boothroyd, D. B., & Johnstone, I. M. (2002). Economic evaluation in long-term clinical trials. *Statistics in Medicine, 21*(19), 2879-2888. doi:10.1002/sim.1292

Kelly, V. L., Barker, H., Field, A. P., Wilson, C., & Reynolds, S. (2010). Can Rachman's indirect pathways be used to un-learn fear? A prospective paradigm to test whether children's fears can be reduced using positive information and modelling a non-anxious response. *Behaviour Research and Therapy, 48*(2), 164-170. doi:10.1016/j.brat.2009.10.002

Kesleman, H. J., Othman, A. R., & Wilcox, R. R. (2016). Generalized Linear Model Analyses for Treatment Group Equality when Data are Non-Normal. *Journal of Modern Applied Statistical Methods, 15*(1), 32-61.

Koller, M. (2016). robustlmm: An R Package for Robust Estimation of Linear Mixed-Effects Models. *Journal of Statistical Software, 75*(6), 1-24. doi:doi:10.18637/jss.v075.i06

Koller, M., & Stahel, W. A. (2011). Sharpening Wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis, 55*(8), 2504-2515. doi:10.1016/j.csda.2011.02.014

Lumley, T., Diehr, P., Emerson, S., & Chen, L. (2002). The importance of the normality assumption in large public health data sets. *Annual Review of Public Health, 23*, 151-169.

Mair, P., Schoenbrodt, F., & Wilcox, R. R. (2017). WRS2: Wilcox robust estimation and testing. R package version (Version 0.9-2). Retrieved from http://cran.r-project.org/package=WRS2

Marks, D. F. (1973). Visual imagery differences in recall of pictures. *British Journal of Psychology, 64*(FEB), 17-24.

Meyer, T. J., Miller, M. L., Metzger, R. L., & Borkovec, T. D. (1990). Development and validation of the penn state worry questionnaire. *Behaviour Research and Therapy, 28*(6), 487-495. doi:10.1016/0005-7967(90)90135-6

Micceri, T. (1989). The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin, 105*(1), 156-166. doi:10.1037//0033-2909.105.1.156

Picardi, A., Battisti, F., De Girolamo, G., Morosini, P., Norcio, B., Bracco, R., & Biondi, M. (2008). Symptom structure of acute mania: A factor study of the 24-item Brief Psychiatric Rating Scale in a national sample of patients hospitalized for a manic episode. *Journal of Affective Disorders, 108*(1-2), 183-189. doi:10.1016/j.jad.2007.09.010

Pratt, J. W. (1964). Robustness of some procedures for 2-sample location problem. *Journal of the American Statistical Association, 59*(307), 665-680. doi:10.2307/2283092

R Core Team. (2016). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.r-project.org/

Ratcliff, R. (1993). Methods for Dealing with Reaction-Time Outliers. *Psychological Bulletin, 114*(3), 510-532.

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software, 48*(2), 1-36. doi:10.18637/jss.v048.i02

Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., Verbeke, T., . . . Maechler, M. (2015). robustbase: Basic Robust Statistics. R package version (Version 0.92-5). Retrieved from http://cran.r-project.org/package=robustbase

Rutter, C. M., & Miglioretti, D. L. (2003). Estimating the accuracy of psychological scales using longitudinal data. *Biostatistics, 4*(1), 97-107.

Steketee, G., Frost, R., Amir, N., Bouvard, M., Carmin, C., Clark, D. A., . . . Obsessive Compulsive Cognitions, W. (2001). Development and initial validation of the obsessive beliefs questionnaire and the interpretation of intrusions inventory. *Behaviour Research and Therapy, 39*(8), 987-1006.

Tangney, J. P., Dearing, R., Wagner, P. E., & Gramzow, R. (2000). *The test of self–conscious affect–3 (TOSCA–3).* Fairfax,VA: George Mason University.

Turner, S. M., Beidel, D. C., & Dancu, C. V. (1996). *Social phobia and anxiety inventory: Manual*. Toronto: Multi–health Systems Inc.

Tyrer, P., Nur, U., Crawford, M., Karlsen, S., McLean, C., Rao, B., & Johnson, T. (2005). The social functioning questionnaire: A rapid and robust measure of perceived functioning. *International Journal of Social Psychiatry, 51*(3), 265-275. doi:10.1177/0020764005057391

Wilcox, R. R. (1998). How many discoveries have been lost by ignoring modern statistical methods? *American Psychologist, 53*(3), 300-314. doi:10.1037//0003-066x.53.3.300

Wilcox, R. R. (2010). *Fundamentals of modern statistical methods: substantially improving power and accuracy* (2nd Ed.). New York: Springer.

Wilcox, R. R. (2016). *Understanding and applying basic statistical methods using R*. Hoboken, New Jersey: John Wiley & Sons.

Wilcox, R. R. (2017). *Introduction to robust estimation and hypothesis testing* (4th ed.). Burlington, MA: Elsevier.

Wilcox, R. R., Carlson, M., Azen, S., & Clark, F. (2013). Avoid lost discoveries, because of violations of standard assumptions, by using modem robust statistical methods. *Journal of Clinical Epidemiology, 66*(3), 319-329. doi:10.1016/j.jclinepi.2012.09.003

Yohai, V. J. (1987). High breakdown-point and high-efficiency robust estimates for regression. *Annals of Statistics, 15*(2), 642-656. doi:10.1214/aos/1176350366

Yuen, K. K. (1974). 2-sample trimmed t for unequal population variances. *Biometrika, 61*(1), 165-170. doi:10.1093/biomet/61.1.165

Zimmerman, M., Chelminski, I., & Posternak, M. (2004). A review of studies of the Hamilton depression rating scale in healthy controls - Implications for the definition of remission in treatment studies of depression. *Journal of Nervous and Mental Disease, 192*(9), 595-601. doi:10.1097/01.nmd.0000138226.22761.39

Comparing two dependent means

Dependent *t*-test

```
> t.test(fieldWide$zThreat, fieldWide$zNone, paired = T)

        Paired t-test

data:  fieldWide$zThreat and fieldWide$zNone
t = 2.8698, df = 42, p-value = 0.006405
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1460035 0.8379076
sample estimates:
mean of the differences
              0.4919556
```

Robust test

```
> yuend(fieldWide$zThreat, fieldWide$zNone, tr = 0.2)
Call:
yuend(x = fieldWide$zThreat, y = fieldWide$zNone, tr = 0.2)

Test statistic: 2.528 (df = 26), p-value = 0.01789

Trimmed mean difference:  0.40456
95 percent confidence interval:
0.0756     0.7335

Explanatory measure of effect size: 0.33
```

Figure 1: Output from an analysis of a one-way repeated measures design (two conditions)

One-way repeated measures design

Repeated measures ANOVA

```
> summary(aov(value ~ InfoType + Error(id/InfoType), data = fieldLong))

Error: id
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 42  51.33   1.222

Error: id:InfoType
          Df Sum Sq Mean Sq F value  Pr(>F)
InfoType   2   8.84   4.420   7.103 0.00141 **
Residuals 84  52.26   0.622
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> pairwise.t.test(fieldLong$value, fieldLong$InfoType, p.adjust.method = "bonferroni", paired = T)

        Pairwise comparisons using paired t tests

data:  fieldLong$value and fieldLong$InfoType

        zNone  zPos
zPos    1.0000 -
zThreat 0.0192 0.0058

P value adjustment method: bonferroni
```

Robust test

```
> rmanovab(fieldLong$value, fieldLong$InfoType, fieldLong$id, tr = 0.2, nboot = 2000)
Call:
rmanovab(y = fieldLong$value, groups = fieldLong$InfoType, blocks = fieldLong$id,
    tr = 0.2, nboot = 2000)

Test statistic: 6.7512
Critical value: 3.2917
Significant:  TRUE

> pairdepb(fieldLong$value, fieldLong$InfoType, fieldLong$id, tr = 0.2, nboot = 2000)
Call:
pairdepb(y = fieldLong$value, groups = fieldLong$InfoType, blocks = fieldLong$id,
    tr = 0.2, nboot = 2000)

                   psihat ci.lower ci.upper     test    crit   sig
zThreat vs. zPos   0.52324  0.14185  0.90463  3.38883 2.47009  TRUE
zThreat vs. zNone  0.40456  0.00926  0.79986  2.52795 2.47009  TRUE
zPos vs. zNone    -0.11868 -0.44481  0.20744 -0.89892 2.47009 FALSE
```

Figure 2: Output from an analysis of a one-way repeated measures design (more than two conditions)

Comparing two independent means

Independent *t*-test

```
> t.test(z ~ Intervention, data = posInfoFBQ)

        Welch Two Sample t-test

data:  z by Intervention
t = 7.1193, df = 53.011, p-value = 2.896e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.035714 1.848215
sample estimates:
     mean in group No Intervention mean in group Positive Information
                         0.6552881                         -0.7866764
```

Robust test

```
> yuenbt(z ~ Intervention, data = posInfoFBQ, tr = 0.2, nboot = 2000)
Call:
yuenbt(formula = z ~ Intervention, data = posInfoFBQ, tr = 0.2,
    nboot = 2000)

Test statistic: 6.0706 (df = NA), p-value = 0

Trimmed mean difference:  1.39458
95 percent confidence interval:
0.9112     1.8779
```

Figure 3: Output from a linear model with one categorical predictor variable (two categories)

One-way independent design

One-way independent ANOVA

```
> summary(aov(z ~ Intervention, data = fbqOnly))
             Df Sum Sq Mean Sq F value   Pr(>F)
Intervention   2  39.02  19.508   30.29 4.31e-11 ***
Residuals    104  66.98   0.644
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> pairwise.t.test(fbqOnly$z, fbqOnly$Intervention, p.adjust.method = "bonferroni")

        Pairwise comparisons using t tests with pooled SD

data:  fbqOnly$z and fbqOnly$Intervention

                     No Intervention Non-Anxious Modelling
Non-Anxious Modelling 0.042          -
Positive Information  3.7e-11        4.8e-06

P value adjustment method: bonferroni
```

Robust test

```
> t1waybt(z ~ Intervention, data = fbqOnly, tr = 0.2, nboot = 2000)
Call:
t1waybt(formula = z ~ Intervention, data = fbqOnly, tr = 0.2,
    nboot = 2000)

Effective number of bootstrap samples was 2000.

Test statistic: 19.8951
p-value: 0
Variance explained 0.66
Effect size 0.812

> mcppb20(z ~ Intervention,  data = fbqOnly, tr = 0.2, nboot = 2000)
Call:
mcppb20(formula = z ~ Intervention, data = fbqOnly, tr = 0.2,
    nboot = 2000)

                                            psihat ci.lower ci.upper p-value
Positive Information vs. Non-Anxious Modelling 0.38249  0.07484  0.75667   0.003
Positive Information vs. No Intervention       1.39458  0.87471  1.88553   0.000
Non-Anxious Modelling vs. No Intervention      1.01209  0.42118  1.56919   0.000
```

Figure 4: Output from a linear model with one categorical predictor variable (more than two categories)

Two-way mixed design

Two-way mixed ANOVA

```
> summary(aov(z ~ Intervention*Measure + Error(id/Measure), data = kellyz))

Error: id
             Df Sum Sq Mean Sq F value   Pr(>F)
Intervention  2  47.54  23.769   32.89 8.55e-12 ***
Residuals   104  75.16   0.723
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: id:Measure
                     Df Sum Sq Mean Sq F value  Pr(>F)
Measure               1   0.00   0.000    0.00 1.00000
Intervention:Measure  2   9.56   4.778    6.23 0.00278 **
Residuals           104  79.75   0.767
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Robust test

```
> bwtrim(z ~ Intervention*Measure, id = id, data = kellyz)
Call:
bwtrim(formula = z ~ Intervention * Measure, id = id, data = kellyz)

                      value p.value
Intervention        42.4514  0.0000
Measure              2.7178  0.1076
Intervention:Measure 6.9346  0.0028
```

Figure 5: Outputs from an analysis of a two-way mixed design

Linear model (regression)

OLS estimation

```
> socAnx.normal<-lm(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
> summary(socAnx.normal)

Call:
lm(formula = socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)

Residuals:
    Min      1Q  Median      3Q     Max
-79.612 -15.805  -0.402  13.671  97.101

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.28871    7.36880   0.718  0.47326
worry        0.43785    0.09484   4.617 4.95e-06 ***
shame        0.05957    0.04145   1.437  0.15128
imagery      0.12155    0.04847   2.508  0.01245 *
obsessive    0.04917    0.01525   3.224  0.00135 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.26 on 507 degrees of freedom
  (47 observations deleted due to missingness)
Multiple R-squared:  0.1229,    Adjusted R-squared:  0.116
F-statistic: 17.76 on 4 and 507 DF,  p-value: 1.181e-13
```

Robust estimation

```
> socAnx.robust<-lmrob(socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
> summary(socAnx.robust)

Call:
lmrob(formula = socAnx ~ worry + shame + imagery + obsessive, data = fieldCH)
 \--> method = "MM"
Residuals:
      Min        1Q    Median        3Q       Max
-77.83290 -15.15220   0.01808  13.88516 101.72303

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.35139    7.88574   0.679 0.497691
worry        0.48445    0.10065   4.813 1.96e-06 ***
shame        0.02594    0.04965   0.522 0.601617
imagery      0.12944    0.04751   2.725 0.006661 **
obsessive    0.05521    0.01585   3.484 0.000537 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Robust residual standard error: 21.74
Multiple R-squared:  0.1414,    Adjusted R-squared:  0.1346
Convergence in 15 IRWLS iterations
```

Figure 6: Output from a linear model with continuous predictor variables

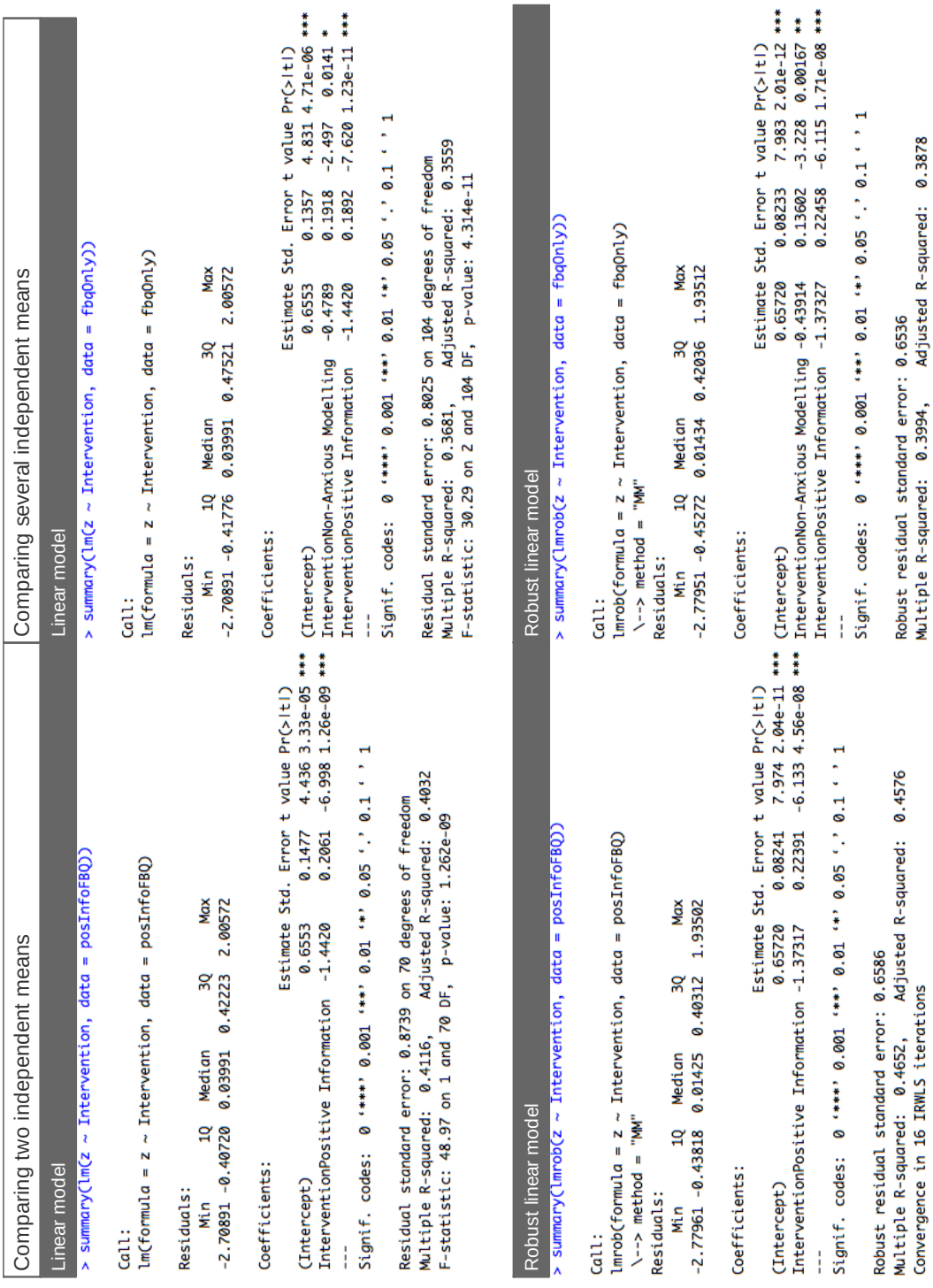


Figure 7: Comparisons between means expressed as linear models using the **lm()** and **lmrob()** functions

Multilevel linear Model

ML estimation

```
> rctLmer<-lmer(Outcome~Group*Time + (Time|ID), data = rctLong, REML = FALSE, na.action = "na.omit")
> summary(rctLmer)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Outcome ~ Group * Time + (Time | ID)
   Data: rctLong

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 ID       (Intercept)  0.03756 0.1938
          Time         0.01320 0.1149   1.00
 Residual             41.97110 6.4785
Number of obs: 600, groups:  ID, 100

Fixed effects:
             Estimate Std. Error t value
(Intercept)   63.83516    0.50085  127.45
GroupTAU       2.35323    0.70777    3.32
Time          -0.86968    0.07217  -12.05
GroupTAU:Time  0.33145    0.10077    3.29

Correlation of Fixed Effects:
            (Intr) GrpTAU Time
GroupTAU    -0.707
Time        -0.649  0.464
GroupTAU:Tm  0.469 -0.664 -0.698
```

Robust estimation

```
> rctRLmer<-rlmer(Outcome~Group*Time + (Time|ID), data = rctLong, REML = FALSE, na.action = "na.omit")
> summary(rctRLmer)
Robust linear mixed model fit by DAStau
Formula: Outcome ~ Group * Time + (Time | ID)
   Data: rctLong

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 ID       (Intercept)  0.00    0.00
          Time         0.00    0.00      NaN
 Residual             41.22    6.42
Number of obs: 600, groups: ID, 100

Fixed effects:
             Estimate Std. Error t value
(Intercept)   63.84739    0.50870  125.51
GroupTAU       2.48596    0.71940    3.46
Time          -0.86349    0.07242  -11.92
GroupTAU:Time  0.31094    0.10242    3.04

Correlation of Fixed Effects:
            (Intr) GrpTAU Time
GroupTAU    -0.707
Time        -0.664  0.470
GroupTAU:Tm  0.470 -0.664 -0.707
```

Figure 8: Edited output from a multilevel linear model (growth model with one two-category fixed effect)

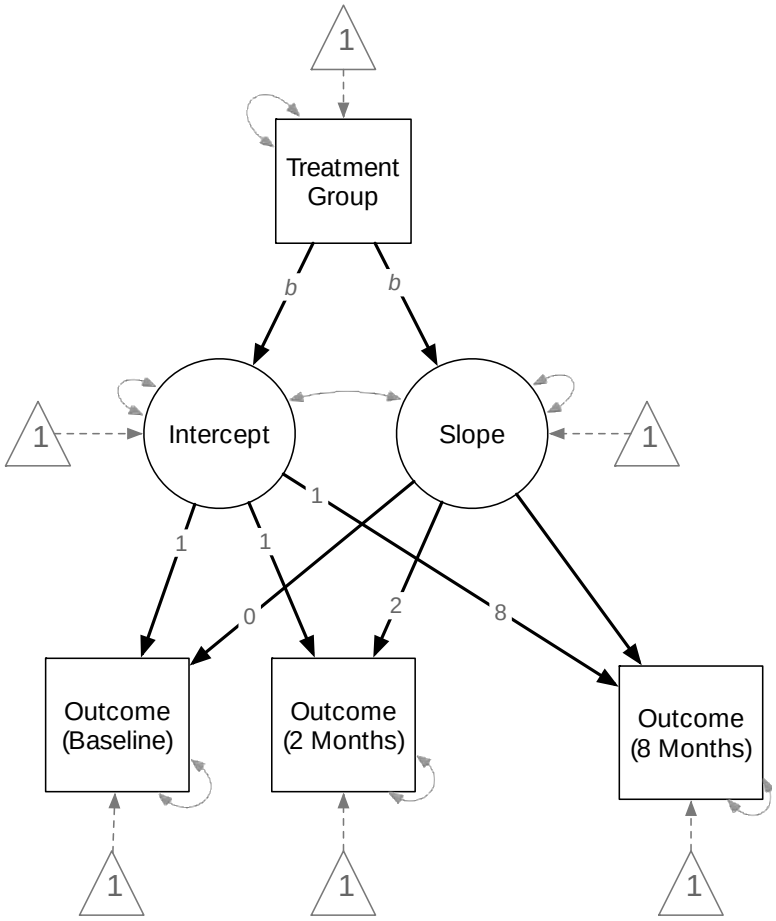
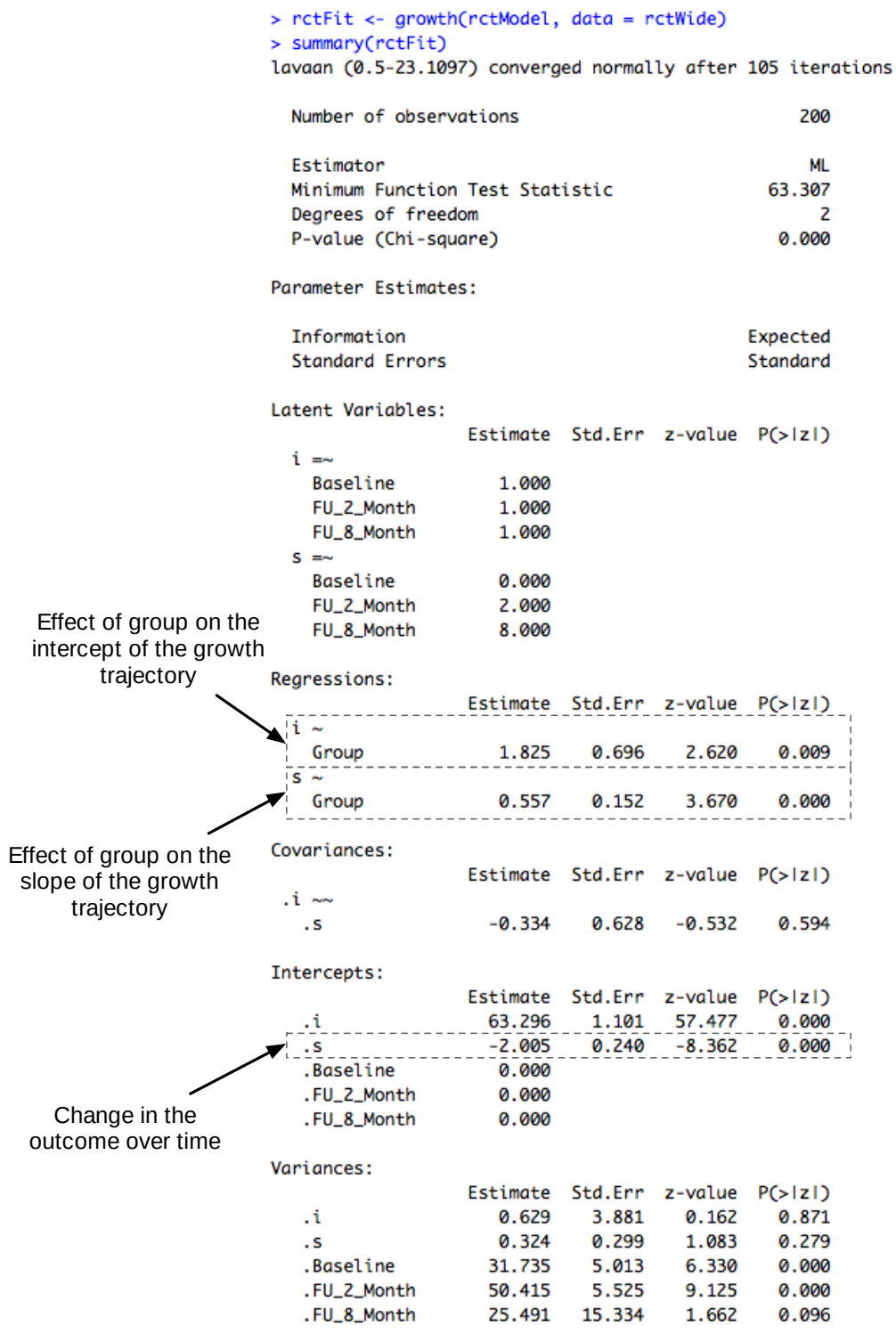


Figure 9: Latent growth model with one categorical predictor




```
> rctFit <- growth(rctModel, data = rctWide)
> summary(rctFit)
lavaan (0.5-23.1097) converged normally after 105 iterations

  Number of observations                           200

  Estimator                                         ML
  Minimum Function Test Statistic               63.307
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  i =~
    Baseline          1.000
    FU_2_Month        1.000
    FU_8_Month        1.000
  s =~
    Baseline          0.000
    FU_2_Month        2.000
    FU_8_Month        8.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  i ~
    Group             1.825    0.696    2.620    0.009
  s ~
    Group             0.557    0.152    3.670    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .i ~~
   .s                -0.334    0.628   -0.532    0.594

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .i                63.296    1.101   57.477    0.000
   .s                -2.005    0.240   -8.362    0.000
   .Baseline          0.000
   .FU_2_Month        0.000
   .FU_8_Month        0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .i                 0.629    3.881    0.162    0.871
   .s                 0.324    0.299    1.083    0.279
   .Baseline         31.735    5.013    6.330    0.000
   .FU_2_Month       50.415    5.525    9.125    0.000
   .FU_8_Month       25.491   15.334    1.662    0.096
```

Figure 10: Output of latent growth model using the maximum likelihood estimator

```
> rctMLR <- growth(rctModel, data=rctWide, estimator = "MLR")
> summary(rctMLR)
lavaan (0.5-22) converged normally after 105 iterations

  Number of observations                           200

  Estimator                                         ML      Robust
  Minimum Function Test Statistic               63.307      60.032
  Degrees of freedom                                 2           2
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  1.055
    for the Yuan-Bentler correction

Parameter Estimates:

  Information                                 Observed
  Standard Errors                    Robust.huber.white

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  i =~
    Baseline          1.000
    FU_2_Month        1.000
    FU_8_Month        1.000
  s =~
    Baseline          0.000
    FU_2_Month        2.000
    FU_8_Month        8.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  i ~
    Group             1.825    0.718    2.541    0.011
  s ~
    Group             0.557    0.152    3.662    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .i ~~
   .s                -0.334    0.656   -0.510    0.610

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .i                63.296    1.186   53.347    0.000
   .s                -2.005    0.244   -8.213    0.000
   .Baseline          0.000
   .FU_2_Month        0.000
   .FU_8_Month        0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .i                 0.629    3.950    0.159    0.874
   .s                 0.324    0.321    1.008    0.313
   .Baseline         31.735    5.996    5.293    0.000
   .FU_2_Month       50.415    5.082    9.920    0.000
   .FU_8_Month       25.491   17.010    1.499    0.134
```

Figure 11: Output of latent growth model using the MLR estimator

Footnotes

[i] The Guass-Markov theorem shows that the OLS estimates of the slope and intercept are essentially a weighted mean of the outcome values. When homoscedasticity is met the OLS estimator minimizes the expected squared error relative to *other weighted means* that might be used. However, there are quite a few robust regression estimators outside of this class that result in smaller standard errors when dealing with an error term that is heavy-tailed, even under homoscedasticity (see Wilcox, 2017). The take home point is that, when using OLS, heteroscedasticity makes things worse relative to many modern robust methods.

[ii] A test statistic is biased if the probability of rejecting the null is not minimized when the null is true.

[iii]You can access many more functions than in official R packages by executing source("http://dornsife.usc.edu/assets/sites/239/docs/Rallfun-v29.txt") in R. (Note that the version number of this document changes regularly so if you receive an error message try replacing v29 in the URL with v30, v31 and so on until the command works.) You can also bootstrap pretty much anything using the **boot()** function in the package *boot* (Canty & Ripley, 2016)

[iv] The package *robustbase* is automatically installed so you need only to reference it with **library()** to access it.

[v] In R you typically refer to variables using the syntax *dataframeName$variableName*, so *fieldWide$zThreat* translates as: the variable called *zThreat* in the dataframe called *fieldWide*.

[vi] To break the first command down, **summary()** prints summary statistics for the model in parenthesis, the model itself is fitted using the **aov()** function which is a special form of the linear model function **lm()**. The main difference is **aov()** returns the table of *F*-statistics that people who use ANOVA are used to seeing, whereas **lm()** returns the specific parameter estimates (and significance tests) and overall fit statistics. The model is specified as *value ~ InfoType + Error(id/InfoType)* which translates as 'I want to predict *value* from the variable *InfoType* plus an error term for that variable that is nested within the variable *id*. Basically then, we are writing out the linear model and the error term tells the function that it is a repeated measures design (because the error term for the predictor variable is nested within cases. Finally, *data = fieldLong* tells the function in which dataframe to find the data.

[vii] Note that R uses a double equals sign to denote 'is equal to'

[viii] Compare this command with that for a one-way repeated measures design (footnote vi). The main difference is that in this command we predict *value* from *InfoType*Measure*. The term *x*y* is a shortcut to including main effects of *x* and *y* and their interaction. Therefore, predicting *value* from *InfoType*Measure* is predicting it from the main effects of *InfoType* and *Measure* as well as their interaction. We could write *value ~ InfoType*Measure* in long form in R as: *value ~ InfoType + Measure + InfoType:Measure*.